TP4 Data Mining: Binary Logistic Regression - Gradient Descent -Newton Method- L2 Regularizer
OBLIGATORY
Individual work

Tuesday 21st April, 2020
deadline: Monday 4th May, 2020, 23:59

1 Objective

The goals of this TP is to optimize the binary logistic regression with an $L2$ norm regularizer using Gradient Descent and Newton's Method to update the weights.

2 Detailed Instructions

This TP is obligatory. You are going to fill a few missing functions in the python script[1] to implement the exercises that we ask. So first of all read and understand the given python script. To run your code you have to run the TP4_main_logistic.ipynb notebook or TP4_main_logistic.py. You are going to use the CIFAR10 data set. This dataset contains 60.000 32x32 color images in 10 different classes (CIFAR-10); For this TP we are going to use olny two classes. To download the data, open a terminal and go to the folder datasets, then execute the script get_datasets.sh:

$$./get_datasets.sh$$

If you have issues to download the data replace datasets folder with this folder.

You have to submit a **formal** report (pdf or using the notebook) and your code.

For this TP the following steps will need to be done:

[1] Part of the given code is based on Stanford's repository

1. The cost function of the logistic regression
   (a) Write down the cost function of the logistic regression on your report
   (b) Derive the gradient of the logistic regression cost with respect to the weights (learning parameters) (Write down the derivations on your report)
   (c) Derive the hessian of the logistic regression cost with respect to the weights (learning parameters) (Write down the derivations on your report)
   (d) Write down the cost function of the logistic regression when a L_2 regularizer is added
   (e) Derive the the gradient of the logistic regressioncost when a L_2 regularizer is added (Write down on your report the final gradient)
   (f) Derive the the hessian of the logistic regressioncost when a L_2 regularizer is added (Write down on your report the final gradient)
2. Based on your derivations, implement the loss function in the logistic regression class and its derivative and Hessian with an L_2 regularizer. **Reminder :** All the formulas and derivations (i.e. how do you get a given formula) must be in the report ! Otherwise the code it's not taken into account.
   (a) Fill the *scores* part in loss function inside the logistic_regression.py
   (b) Fill the *loss* part in loss function inside the logistic_regression.py
   (c) Fill the *gradient* part in loss function inside the logistic_regression.py
   (d) Fill the *hessian* part in loss function inside the logistic_regression.py
3. Fill the missing part of predict function inside the logistic_regression.py
4. Gradient Descent and Newton's Method updates
   (a) Fill the missing parts of the train() method inside the classifier.py script to update the weights using Gradient Descent and Newton's Method
5.
6. Fill the missing part in TP4_main_logistic.ipynb notebook or TP4_main_logistic.py and train your classifier for different learning rates and regularization strengths using Gradient Descent and Newton's Method to update the weights. Count the computation time for each case/combination.
   (a) Comment **in details** how the different learning rates and regularization strengths influence the performance of the classifier. Which is the effect of very large/small regularizers? How the learning rate change the prediction? The computational time that each update method needs? etc..

(b) Find the best hyperparameters by tuning on the validation set using gradient descent and newton method to update the weights.

   i. Plot the loss history of the best model using Gradient Descent and of the best model using Newton's Method on test set. Based on the results of the best models compare the two update methods, comment on the number of the iterations that each update method needs and the computational time that each one is needed, etc.

   ii. Evaluate the best model using Gradiend Descent and the best model using Newton's Method on test set and comment on the result.

3 Reminder

- The L2 regularizer is also called Ridge Regression. It adds "squared magnitude" of weights as penalty term to the cost function. If the regularizer parameter is zero then you can imagine we get back to the cost function.